4

SRL-0006-TM AR-005-351

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
SALISBURY

SURVEILLANCE RESEARCH LABORATORY

SOUTH AUSTRALIA

TECHNICAL MEMORANDUM

SRL-0006-TM

THE FAST HARTLEY TRANSFORM AS AN ALTERNATIVE TO THE FAST FOURIER TRANSFORM

F. PICCININ

Technical Memoranda are of a tentative nature, representing the views of the author(s), and do not necessarily carry the authority of the Laboratory.

Approved for Public Release

FEBRUARY 1988

COPY No. 28

The official documents produced by the Laboratories of the Defence Science and Technology *Organisation Salisbury are issued in one of five categories:* Reports, Technical Reports, Technical Memoranda, Manuals and Specifications. The purpose of the latter two categories is self-evident, with the other three categories being used for the following purposes:

Reports	:	documents prepared for managerial purposes.
Technical Reports	:	records of scientific and technical work of a permanent value intended for other scientists and technologists working in the field.
Technical Memoranda	:	intended primarily for disseminating information within the DSTO. They are usually tentative in nature and reflect the personal views of the author.

AR-005-351

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
SURVEILLANCE RESEARCH LABORATORY

TECHNICAL MEMORANDUM

SRL-0006-TM

THE FAST HARTLEY TRANSFORM AS AN ALTERNATIVE TO THE FAST FOURIER TRANSFORM

F. PICCININ

SUMMARY

A comparison is made between the Discrete Hartley Transform and Discrete Fourier Transform algorithms. The Fast Hartley Transform is examined as an alternative to the Fast Fourier Transform for signal processing in the Jindalee Over The Horizon Radar project.

POSTAL ADDRESS: Director, Surveillance Research Laboratory,
Box 1650, PO, Salisbury, South Australia, 5108

Contents

1 INTRODUCTION

Although the Discrete Hartley Transform (DHT) has been around for many years(ref.1,2), considerable new interest in this transform has been generated recently. This renewed interest is a result of the discovery of a Fast Hartley Transform (FHT) algorithm(ref.3). In common with the Fast Fourier Transform (FFT), the FHT algorithm computes the DHT of a data sequence of N elements in a time proportional to $N \log_2 N$. Early work(ref.3) indicates the FHT is as fast or faster than the FFT, inferring the FHT is a more efficient substitute for the FFT in areas such as spectral analysis, digital processing, and convolution.

The signal processing scheme for the Jindalee Over The Horizon Radar (OTHR) employs the FFT for range processing, digital beamforming, and Doppler processing. The FFT constitutes a significant portion of the total processing load. Future developments in operational OTH radars for Australia will lead to an increased range processing load, which relies almost exclusively on the FFT, hence a more efficient algorithm is of interest.

The definition of the DHT is given in Section 2, along with a summary of its properties. In Section 3 the fast DHT and DFT transforms are described and comparison made of the number of processing steps required for the FFT and FHT algorithms. Section 4 establishes the suitability of these two algorithms for Jindalee signal processing, and conclusions are made in Section 5.

2 THE HARTLEY TRANSFORM

Consider a sequence of N real numbers x_n for $n = 0, 1, ...N - 1$. The Discrete Hartley Transform of this sequence is defined(ref.3) as

$$H_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n \left[cos(\frac{2\pi nk}{N}) + sin(\frac{2\pi nk}{N})\right] \quad (1)$$

$$k = 0, 1, ...N - 1.$$

This is often written as

$$H_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n cas(\frac{2\pi nk}{N}) \quad (2)$$

$$k = 0, 1, ...N - 1$$

where $cas(\theta) = cos(\theta) + sin(\theta)$.

The inverse DHT of a sequence of N real numbers H_k for $k = 0, 1, ...N - 1$ is given by

$$x_n = \sum_{k=0}^{N-1} H_k[cas(\frac{2\pi nk}{N})] \quad (3)$$

$$n = 0, 1, ...N - 1.$$

The DHT has a number of interesting properties, that can be more readily understood by comparison with the Discrete Fourier Transform (DFT). The corresponding expressions for the DFT and inverse DFT are

$$F_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n[cos(\frac{2\pi nk}{N}) - j\, sin(\frac{2\pi nk}{N})] \quad (4)$$

$$k = 0, 1, \ldots N-1.$$

$$x_n = \sum_{k=0}^{N-1} F_k[\cos(\frac{2\pi nk}{N}) + j\,\sin(\frac{2\pi nk}{N})] \tag{5}$$

$$n = 0, 1, \ldots N-1.$$

From equations (2) and (3) we see that the forward and inverse DHTs are identical (apart from a scaling factor). This can be of advantage on limited memory machines, requiring only one algorithm be stored in program memory, rather than the two required for the (inverse) DFT. Equations (4) and (5) show that the forward and inverse DFTs differ by a sign change of the imaginary part. Also, the DHT uses real arithmetic only, while the DFT requires complex arithmetic. The absence of complex arithmetic gives the DHT the appearance of being simpler. It will be shown, however, that both the DHT and DFT are of similar complexity.

It must be emphasised that the DHT and DFT are distinct transforms. Both offer an alternative way of representing the same data. The DFT representation of the data sequence x_n for $n = 0, 1, \ldots N-1$ gives us amplitude and phase information on sinusoidal frequencies present in x_n. The DHT gives the same information, but in a slightly modified form.

From equations (1) and (4) it is clear that the Hartley Transform H_k is not the same as the Fourier Transform F_k. As it is the Fourier Transform that we generally desire, the Hartley Transform is only of use if we can readily derive the Fourier Transform from it.

The DFT may be derived from the DHT as follows. Remembering that *cos* is an even function and *sin* is an odd function, inspection of equations (1) and (4) reveals that, for real x_n with $n = 0, 1, \ldots N-1$, the even part of the DHT is equivalent to the real part of the DFT, and the odd part of the DHT is equivalent to the negative of the imaginary part of the DFT. Thus the DFT can be derived from the DHT by $\frac{N}{2}$ add operations and $\frac{N}{2}$ subtract operations (ignoring scaling by 2). Noting that $H_0 = H_N$ and $F_0 = F_N$, mathematically we can express the required relationships as

$$\mathcal{R}eal\{F_k\} = \frac{1}{2}[H_k + H_{N-k}] \tag{6}$$

$$n = 0, 1, \ldots \frac{N}{2} - 1$$

with $\mathcal{R}eal\{F_k\}$ symmetrical about $F_{N/2}$ ie $\mathcal{R}\{F_k\} = \mathcal{R}\{F_{N-k}\}$.

$$\mathcal{I}mag\{F_k\} = -\frac{1}{2}[H_k - H_{N-k}] \tag{7}$$

$$n = 0, 1, \ldots \frac{N}{2} - 1$$

with $\mathcal{I}mag\{F_k\}$ anti-symmetric about $F_{N/2}$ ie $\mathcal{I}\{F_k\} = -\mathcal{I}\{F_{N-k}\}$.

From the above it should be evident that for real x_n with $n = 0, 1, \ldots N-1$, to determine the DFT F_k for $k = 0, 1, \ldots N-1$ via the DHT, it is necessary to calculate the Hartley Transform H_k for $k = 0, 1, \ldots N-1$. There exist redundancies, however, in the DFT. It can be readily seen that $F_k = F^*_{N-k}$, so it is only necessary to determine F_k for $k = 0, 1, \ldots \frac{N}{2} - 1$. ie: only half of the Fourier Transform need be calculated for real x_n. This redundancy balances with the DHT's absence of complex arithmetic, making both transforms of similar complexity (in terms of the number of arithmetic steps required).

The DFT can be performed on a sequence of complex data (ie: x_n complex in (4)) and the transform will be of the same length and generally will also be complex. In contrast, the DHT can only be performed on a real data sequence (x_n real in (2)), the transform also being a real data sequence of the same length. We have the problem then: how do we use the Hartley algorithm to transform a complex sequence?

We can determine the Fourier Transform of a complex data sequence via the Hartley Transform by separately transforming the real and imaginary parts of the complex sequence, and then recombining these transforms. These steps are illustrated mathematically below.

Consider the complex sequence $z_n = x_n + jy_n$ for $n = 0, 1, ...N-1$ where x_n and y_n are both real sequences. Denoting the Fourier Transform operator by $F\{\ \}$ we have, by linearity,

$$\begin{aligned} Z_k = F\{z_n\} &= F\{x_n + jy_n\} \\ &= F\{x_n\} + jF\{y_n\} \\ &= X_k + jY_k \\ k &= 0, 1, ...N-1 \end{aligned} \tag{8}$$

where X_k, Y_k, and Z_k are the Fourier Transforms of x_n, y_n, and z_n respectively. The DHT can be used to compute X_k and Y_k from x_n and y_n in the manner described above. Thus with little extra complexity, the DHT can be used to compute the DFT for real or complex data (complex data requiring two distinct Hartley Transforms be performed).

3 PRINCIPLES OF FAST DHT AND DFT ALGORITHMS

While there exist many applications for the Discrete Fourier Transform (eg spectral analysis, correlation, convolution), computing the DFT via the definition given in Section 2 requires a considerable amount of processing. To directly transform a sequence of length N requires a number of arithmetic operations of order N^2 ie doubling the length of the original sequence results in a four-fold increase in the processing load. For large N this method becomes impractical, requiring enormous processing power. In 1965 a fast alternative method of determining the DFT was reported(ref.4). This Fast Fourier Transform could compute the DFT in a time proportional to $N \log_2 N$, making it far more suitable than direct calculation when transforming sequences containing many points.

The development of the Hartley Transform has proceeded in an analogous way. The Discrete Hartley Transform was first defined in 1942(ref.1). Calculation via the definition, given in Section 2, executes in a time proportional to N^2. As with the FFT, a Fast Hartley Transform has been defined(ref.3). This FHT algorithm, like the FFT, also executes in a time proportional to $N \log_2 N$, generating considerable interest since its discovery by Bracewell in 1984.

It is instructive to analyse the way in which these fast transforms work. We will look at the radix-2 FHT and FFT algorithms. Although not the most efficient fast algorithms, the radix-2 transforms are the most widely understood fast method of transforming a sequence of length N, with $N = 2^i$: i integer.

3.1 THE RADIX-2 FFT

To determine the Fourier Transform of x_n, $n = 0, 1, \ldots N-1$ and $N = 2^i$;

let

$$y_n = x_{2n}$$

$$z_n = x_{2n+1}$$

$$n = 0, 1, \ldots \frac{N}{2} - 1.$$

The sequences y_n and z_n are each of length $\frac{N}{2}$, and have transforms

$$Y_k = \sum_{n=0}^{\frac{N}{2}-1} y_n e^{-j\frac{4\pi nk}{N}} \tag{9}$$

$$Z_k = \sum_{n=0}^{\frac{N}{2}-1} z_n e^{-j\frac{4\pi nk}{N}} \tag{10}$$

$$k = 0, 1, \ldots \frac{N}{2} - 1.$$

The transform that we seek is X_k,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi nk}{N}} \tag{11}$$

$$k = 0, 1, \ldots N-1.$$

Expression (11) can be manipulated to give

$$X_k = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi(2n)k}{N}} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi(2n+1)k}{N}}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} y_n e^{-j\frac{4\pi nk}{N}} + e^{-j\frac{2\pi k}{N}} \sum_{n=0}^{\frac{N}{2}-1} z_n e^{-j\frac{4\pi nk}{N}}$$

so that

$$X_k = Y_k + e^{-j\frac{2\pi k}{N}} Z_k. \tag{12}$$

Also

$$X_{k+\frac{N}{2}} = Y_k + e^{-j\pi} e^{-j\frac{2\pi k}{N}} Z_k$$

or

$$X_{k+\frac{N}{2}} = Y_k - e^{-j\frac{2\pi k}{N}} Z_k. \tag{13}$$

Expressions (12) and (13) are often written as

$$X_k = Y_k + W^k Z_k \tag{14}$$

$$X_{k+\frac{N}{2}} = Y_k - W^k Z_k \tag{15}$$

with

$$W^k = e^{-j\frac{2\pi k}{N}}.$$

The pair of equations (14) and (15) is often referred to as the FFT kernel or butterfly operation, as it represents the fundamental building block of the radix-2 FFT algorithm. The butterfly operation requires $\frac{N}{2}$ complex multiplications and N complex additions to compute the sequence X_k for $k = 0, 1, 2...N-1$, from the sequences Y_k and Z_k, $k = 0, 1, 2...\frac{N}{2}$.

The term 'radix-2' is due to X_k being determined from the *two* transforms Y_k and Z_k. A large number of different radix algorithms have been tried, including radix-4 and radix-8. Interestingly, one of the fastest FFT algorithms, the Split-Radix, uses a hybrid scheme where the length N DFT is computed from a length $\frac{N}{2}$ plus two length $\frac{N}{4}$ DFTs.

The principle of the radix 2 FFT can be described as :

1. generate length N DFT from 2 length $\frac{N}{2}$ DFTs
2. generate each length $\frac{N}{2}$ DFT from 2 length $\frac{N}{4}$ DFTs

.

.

i. generate each length 2 DFT from 2 length 1 DFTs,
-each length 1 DFT is equal to itself.

Each of the above i steps ($i = \log_2 N$) requires $\frac{N}{2}$ complex multiplications and N complex additions. Thus the length N FFT requires $\frac{N}{2}\log_2 N$ complex multiplications and $N\log_2 N$ complex additions.

3.2 THE RADIX-2 FHT

The development of the Fast Hartley Transform proceeds in a similar way to the FFT. Suppose we wish to determine the Hartley Transform of a real sequence x_n for $n = 0, 1, 2...N-1$ and $N = 2^i$ with i:integer.
Let

$$y_n = x_{2n}$$

$$z_n = x_{2n+1}$$

$$n = 0, 1, ...\frac{N}{2} - 1.$$

The real sequences y_n and z_n are each of length $\frac{N}{2}$ and have transforms

$$Y_k = \sum_{n=0}^{\frac{N}{2}-1} y_n cas(\frac{4\pi nk}{N})$$

$$Z_k = \sum_{n=0}^{\frac{N}{2}-1} z_n cas(\frac{4\pi nk}{N})$$

$$k = 0, 1, ...\frac{N}{2} - 1.$$

We seek the transform X_k,

$$X_k = \sum_{n=0}^{N-1} x_n[cos(\frac{2\pi nk}{N}) + sin(\frac{2\pi nk}{N})].$$

Expanding gives

$$X_k = \sum_{n=0}^{\frac{N}{2}-1} y_n cas(\frac{4\pi nk}{N}) + \sum_{n=0}^{\frac{N}{2}-1} z_n[cos(\frac{4\pi nk}{N} + \frac{2\pi k}{N}) + sin(\frac{4\pi nk}{N} + \frac{2\pi k}{N})].$$

This can be further expanded to give

$$X_k = \sum_{n=0}^{\frac{N}{2}-1} y_n cas(\frac{4\pi nk}{N}) + \sum_{n=0}^{\frac{N}{2}-1} z_n[cos(\frac{4\pi nk}{N})cos(\frac{2\pi k}{N}) - sin(\frac{4\pi nk}{N})sin(\frac{2\pi k}{N})$$

$$+sin(\frac{4\pi nk}{N})cos(\frac{2\pi k}{N}) + cos(\frac{4\pi nk}{N})sin(\frac{2\pi k}{N})].$$

Hence

$$X_k = Y_k + [cos(\frac{2\pi k}{N})Z_k + sin(\frac{2\pi k}{N})Z_{N-k}] \tag{16}$$

and

$$X_{k+\frac{N}{2}} = Y_k - [cos(\frac{2\pi k}{N})Z_k + sin(\frac{2\pi k}{N})Z_{N-k}]. \tag{17}$$

Equations (16) and (17) form the basis of the FHT algorithm. As with the FFT, the FHT calculates the length N transform from two length $\frac{N}{2}$ transforms. For the radix-2 algorithms considered here, both the FHT and FFT require the same number of butterflies.

We see that the FHT butterfly requires two real multiplications and three real additions, where the FFT butterfly requires one complex multiplication and two complex additions. Reasoning in the same way as for the FFT, it can be shown that to calculate an N point FHT requires $N\log_2 N$ real multiplications and $\frac{3}{2}N\log_2 N$ real additions in the form of butterflies. As described in section 2, further operations of order N will also be required to compute the DFT from this Hartley Transform.

Comparing the execution speed of the FHT and FFT is not straight forward. The FFT butterfly requires the equivalent of four real multiplies and 6 real adds, twice as many as for the FHT butterfly, but the FHT can only be used for real data. Generally twice as many FHT butterflies will be required for a complex transform, so that both the FHT and FFT require the same number of real arithmetic operations from butterflies alone. Any advantages that one may have over the other will be of order N. For large N this difference of order N will become relatively less significant, when compared with the total number of arithmetic operations (which is of order $N\log_2 N$). The issue of execution speed comparison is complicated further by the existence of more efficient FFT and FHT algorithms. In particular, an alternative real valued FFT (RFFT) exists(ref.5) that is shown to be faster than any known FHT algorithm.

It is easier to compare the FFT and FHT with a specific application in mind. The next section looks at the specific application of the FHT and FFT to Jindalee signal processing, where purpose-built computer hardware is used to increase processing throughput.

4 APPLICATION OF FHT/FFT TO JINDALEE SIGNAL PROCESSING

This section looks at the suitability of the FHT/FFT for current Jindalee signal processing, and also the implications for hardware design in future Jindalee radars. The following comparisons assume a radix-2 FHT and FFT. This is justified as it has been shown(ref.6,7) that due to the similarity of the algorithms, any optimisation applied to one can also be applied to the other with the same speed improvement.

Signal processing for the Jindalee OTHR(ref.8) relies heavily on the FFT. It is used for ranging, where radar returns are separated into range bins; digital beamforming, where radar returns are separated into azimuthal 'finger beams'; and Doppler processing, where targets' radial speeds allows them to be separated from the large land/sea backscatter return.

In order to meet the required FFT load, Arithmetic Oriented (ARO) array processors were designed and built(ref.9) at ERL. The hardware of the ARO processor is optimised around the FFT butterfly operation, with a hardware multiplier and two hardware adders operating in parallel (the radix-2 FFT butterfly requires a complex multiply and two complex adds). By employing a degree of pipe-lining, each of these arithmetic units can perform a complex operation in 240 ns, or a real operation in 320 ns. The combination of these parallel arithmetic units allows a complex butterfly to be computed in 240 ns (three microcycles of the arithmetic processor (AP)). It actually takes 27 AP microcycles to perform a butterfly from start to finish. The pipe-lining, however, allows a butterfly to be completed every 240 ns.

It can be readily demonstrated that the ARO processor is not suited to calculating the FHT efficiently. Consider the FHT butterfly, requiring two real multiplications and three real adds. The butterfly could not be performed in less than 480 ns (assuming the two real multiplications are performed as complex operations) and the FHT is capable of transforming real data only. Jindalee signal processing requires a complex transform be done ie two separate Hartley Transforms are required (as explained in Section 2). It is apparent the FFT algorithm will execute approximately four times faster than the FHT on the ARO processor (2 FHTs against 1 FFT and butterfly time at least twice as long for the FHT). Clearly it would not be practical to implement the FHT on the ARO.

Future Jindalee radars will have an increased range and Doppler processing FFT load. The digital beamforming load will also increase, but this is not likely to be done via FFT. For this increased signal processing load, faster arithmetic processors will be required. We now look briefly at the suitability of designing a new processor around the FHT rather than the FFT.

To perform a real or complex Fourier Transform requires almost the same number of real operations for both the FHT and FFT. Also, due to the similarity of the two fast algorithms, neither appears to be more suited than the other in terms of ease of arithmetic hardware design. It is therefore concluded that the Fast Hartley Transform offers no advantages over the Fast Fourier Transform for the design of any future signal processor.

To increase FFT through-put there are other areas that can be addressed. The ARO processor implements a radix-2 FFT. There exist FFT algorithms, such as the split-radix, that execute in about half the time(ref.6). Discussion with colleagues indicates a hardware limitation prevents the use of a faster FFT on the ARO.

A further limitation of the ARO processor is the lack of hardware 'bit-reversal'. Bit-reversal is required to unscramble data prior to or after performing an FFT. This bit-reversal is currently done in software and considerably slows the FFT routine. These hardware inadequacies, plus factors such as vector set-up times, become more dominant for transforms of short data sequences, and should be taken into account when designing a new processor.

5 CONCLUSION

Despite much recent interest, there appears to be no significant benefit in using the Fast Hartley Transform in place of the Fast Fourier Transform. The FHT requires a comparable number of steps to execute and is of comparable complexity to the FFT. The FHT does have the advantage that the forward and inverse transforms are the same, but this is only of advantage on a limited memory machine.

For any future arithmetic processor, improved FFT performance can be achieved by addressing the ARO hardware limitations. In particular

— Base the arithmetic hardware around the split-radix rather than radix-2 butterfly (or perhaps design the hardware so that any new algorithms can be readily microcoded in the future)

— Implement hardware bit-reversal

— Reduce vector set-up times

Bibliography

[1] R.V.L. Hartley, "A more symmetrical Fourier analysis applied to transmission problems," Proc. IRE, Vol. 30, pp. 144-150, March 1942.

[2] R.N. Bracewell, "The Discrete Hartley Transform," J. Opt. Soc. Amer., Vol. 73, pp. 1832-1835, December 1983.

[3] R.N. Bracewell, "The Fast Hartley Transform," Proc IEEE, Vol. 72, no. 8, pp. 1010-1018, August 1984.

[4] J.W. Cooley and J.W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput., Vol. 19, pp. 297-301, April 1965.

[5] H.V. Sorensen, D.L. Jones, M.T. Heideman and C.S. Burrus, "Real-Valued Fast Fourier Transform Algorithms," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-35, pp. 849-863, June 1987.

[6] G.E.J. Bold, "A Comparison of the time involved in computing fast Hartley and Fast Fourier Transforms," Proc. IEEE, Vol. 73, pp. 1863-1864, December 1985.

[7] H.V. Sorensen, D.L. Jones, C.S. Burrus and M.T. Heideman, "On computing the discrete Hartley transform," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-33, pp. 1231-1238, October 1985.

[8] M.L. Lees "A Signal Processing Scheme for HF Radar," Electronics Research Laboratory, *Technical Report ERL-0382-TR*, September 1986, Confidential

[9] P.C. Drewer and J. Ziukelis, "The ARO array processor," Proc. of Aust. Comp. Eng. Conf No. 1, Newcastle 1983.

DISTRIBUTION

DEPARTMENT OF DEFENCE

	COPY No.
Defence Science and Technology Organisation	
Chief Defence Scientist	1
Assistant Chief Defence Scientist (Policy)	
Assistant Chief Defence Scientist (Operations)	
Director General External Relations	
Director General Science Technology Programs	
Counsellor, Defence Science, London	Cnt Sht Only
Counsellor, Defence Science, Washington	Cnt Sht Only
Superintendent, Analytical Studies	2
Superintendent, Major Projects	3
Surveillance Research Laboratory	
Director, Surveillance Research Laboratory	4
Superintendent, High Frequency Radar Division	5
Senior Principal Research Scientist, HF Radar	6
Principal Officer, Ionospheric Effects Group	7
Principal Officer, Radar Processing and Tracking Group	8
Principal Officer, Radar Technology and Systems Group	9
Principal Officer, Radar Engineering Group	10

Dr. S.B. Colegrove, Radar Processing and Tracking Group	11
Dr. M.D.E. Turley, Radar Processing and Tracking Group	13
Mr. F. Piccinin, Radar Processing and Tracking Group	14
Dr. D.J. Netherway, Radar Processing and Tracking Group	15
Air Force	
Air Force Scientific Adviser	16
Director Joint Intelligence Organisation (DSTI)	17
Libraries and Information Services	
Librarian, Technical Reports Section, Defence Library, Campbell Park	18
OIC Document Exchange Centre, Defence Information Service, for	
Microfiche copying then destruction	19
United Kingdom, Defence Research Information Center	20 - 21
United States, Defence Technical Information Center	22 - 33
Canada, Director Scientific Information Services	34
New Zealand, Ministry of Defence	35
National Library of Australia	36
Library, DSD Melbourne	37
Main Library, DSTO, Salisbury	38 - 39
Defence Communications System Division	
(Attention: General Manager)	40
(Attention: Dr. P.S. Whitham, SRS-1, OTHR Project Office)	41
Spares	42 - 43

DOCUMENT CONTROL DATA SHEET

Security classification of this page : UNCLASSIFIED

1 DOCUMENT NUMBERS

AR Number : AR-005-351

Series Number : SRL-0006-TM

Other Numbers :

2 SECURITY CLASSIFICATION

a. Complete Document : Unclassified

b. Title in Isolation : Unclassified

c. Summary in Isolation : Unclassified

3 DOWNGRADING / DELIMITING INSTRUCTIONS

Limitation to be reviewed in February 1991

4 TITLE

THE FAST HARTLEY TRANSFORM AS AN ALTERNATIVE TO THE FAST FOURIER TRANSFORM

5 PERSONAL AUTHOR (S)

F. Piccinin

6 DOCUMENT DATE

February 1988

7 7.1 TOTAL NUMBER OF PAGES 9

7.2 NUMBER OF REFERENCES 9

8 8.1 CORPORATE AUTHOR (S)

Surveillance Research Laboratory

8.2 DOCUMENT SERIES and NUMBER

Technical Memorandum 0006

9 REFERENCE NUMBERS

a. Task :

b. Sponsoring Agency :

10 COST CODE

142/787954

11 IMPRINT (Publishing organisation)

Defence Science and Technology Organisation Salisbury

12 COMPUTER PROGRAM (S) (Title (s) and language (s))

13 RELEASE LIMITATIONS (of the document)

Approved for Public Release

Security classification of this page : UNCLASSIFIED

Security classification of this page : UNCLASSIFIED

14 ANNOUNCEMENT LIMITATIONS (of the information on these pages)

No limitation

15 DESCRIPTORS

a. EJC Thesaurus Terms	Algorithms Fast Fourier Transforms Signal processing Over The Horizon Radar	
b. Non - Thesaurus Terms	Fast Hartley Transforms	

16 COSATI CODES

0072B

17 SUMMARY OR ABSTRACT
(if this is security classified, the announcement of this report will be similarly classified)

A comparison is made between the Discrete Hartley Transform and Discrete Fourier Transform algorithms. The Fast Hartley Transform is examined as an alternative to the Fast Fourier Transform for signal processing in the Jindalee Over The Horizon Radar project.

Security classification of this page : UNCLASSIFIED